第27卷 第6期 工 程 数 学 学 报 Vol. 27 No. 6
2010年12月 CHINESE JOURNAL OF ENGINEERING MATHEMATICS Dec. 2010

文章编号:1005-3085(2010)06-0967-08

非线性混合整数规划问题的改进差分进化算法*

刘俊梅[1,2]， 高岳林[2]

(1- 中国矿业大学银川学院基础部数学教研室，银川 750011;
2- 北方民族大学信息与系统科学研究所，银川 750021)

摘 要: 针对非线性混合整数规划问题，本文采用非固定多段映射罚函数法处理约束条件、用混合整数编码技术处理连续变量和整数变量，并在基本差分进化算法中加入一种新型的凸组合变异算子和一种指数递增交叉算子，由此构造出了一种求解非线性混合整数规划问题的改进差分进化算法。实验表明，所提出的算法全局收敛速度快，精度高，鲁棒性强。

关键词: 全局优化；非线性混合整数规划；非固定多段映射罚函数；差分进化算法

分类号: **AMS**(2000) 92B99 **中图分类号**: TP18 **文献标识码**: A

1 引言

混合整数非线性规划问题 (MINP) 广泛应用于机械、化工、资源管理、生产调度、生物、军事等领域。求解 MINP 的传统方法有分支定界法、广义 Benders 分解法等。由于同时含有实数变量和整数变量，MINP 是一类 NP 完全问题，随着变量维数的增加，计算量会急剧增大，从而使这些方法存在很大的局限。目前进化计算方法广泛用于解决约束优化问题，且被证明很有效[1-4]。近年来，有许多学者将进化算法，如遗传算法 (GA)[1-3]、模拟退火算法 (SA)[4]、进化规划 (EP)[5]、粒子群优化算法 (PSO)[6,7] 和差分进化算法 (DE) 等用于求解 MINP 问题。这些方法一般都取得了满意的效果[6]。差分进化算法是 Storn 和 Price 于 1995 年共同提出的一种采用浮点矢量编码，在连续空间中进行启发式随机搜索的优化算法[8]。差分进化算法作为一种性能卓越的优化算法正受到日益关注，其应用领域也越来越广。为了将 DE 用于 MINP 问题，文献 [9] 对 DE 的变异操作进行了改进，提出了一种对于整数变量直接在整数空间进行优化计算的改进差分进化算法。文献 [10] 对变异矢量采用四舍五入方法进行取整，使之适合于 0-1 整数规划和整数只包括 0, 1 的优化问题。对于其它 MINP 问题，同时进行向上取整和向下取整运算，得到两个试验向量，从而扩大了寻优空间，有利于提高算法搜索到最优解的鲁棒性，取得了满意的效果。

本文在文献 [10] 的基础上，采用非固定多段映射罚函数法处理问题的约束条件，用混合整数编码技术处理连续变量和整数变量，结合差分进化算法两种不同变异方式的特点，给出一种线性递减加权因子的凸组合[11,12] 变异方案，并采用指数递增交叉算子提高算法的全局搜索能力和收敛速率，用四个经典测试函数对改进算法进行了测试，实验结果表明了所提出的改进 DE (Modified DE, MDE) 算法用于求解 MINP 问题的有效性。

收稿日期: 2008-12-26. **作者简介**: 刘俊梅 (1980年9月生)，女，硕士. 研究方向：最优化理论方法及其应用，智能计算及其应用.
***基金项目**: 国家自然科学基金 (60962006).

2 MINP 问题描述

一般混合整数非线性规划问题可描述为

$$\begin{cases} \min & f(x)=(x,y) \\ \text{s.t.} & g_i(x,y)\le 0, \quad i=1,2,\cdots,m, \\ & h_j(x,y)=0, \quad j=1,2,\cdots,l, \\ & x^L\le x\le x^U, \quad y^L\le y\le y^U, \end{cases} \tag{1}$$

式 (1) 中 x 为连续矢量，y 为整数矢量，(x^L,x^U) 和 (y^L,y^U) 分别为相应决策矢量的下界和上界；$g_i(x,y)\le 0$ 为不等式约束条件，$h_j(x,y)=0$ 为等式约束条件。这些约束条件通常都是非线性的，因而 MINP 问题一般难以求解。对于等式约束，可以令 $h_j(x,y)\le 0$ 和 $h_j(x,y)\ge 0$ 化为不等式约束来处理。通常所构造的广义目标函数具有如下形式

$$F(x,y)=f(x,y)+\delta(t)H(x,y), \tag{2}$$

其中 $f(x)=(x,y)$ 表示原目标函数，$\delta(t)H(x,y)$ 称为惩罚项，$\delta(t)$ 表示惩罚力度，$H(x,y)$ 为惩罚因子。式 (2) 中如果在求解约束优化问题 (COP) 的整个过程中 $\delta(t)$ 固定不变，则称为固定罚函数法，相反则称为非固定罚函数法。固定罚函数法也称为精确罚函数法，在整个求解过程中只需计算一个目标函数，但惩罚力度 $\delta(t)$ 难以选择。如果太小，则收敛减慢，甚至出现不收敛的情形；如果太大，则会在可行域边界使目标函数产生病态，给计算带来困难[8]。而非固定罚函数法则在求解的过程中，$\delta(t)$ 随迭代次数 t 是变化的，将约束问题转化为一系列无约束问题来求解。Parsopoulos 和 Vrahatis 用 PSO 算法求解 COP 时，提出了一种称之为非固定多段映射罚函数法[12-14]，对约束条件进行处理。本文在文献 [10] 的基础上，对分段映射函数进行了适当修正。设

$$H(x,y)=\sum_{i=1}^{m+2l}\theta\big(q_i(x,y)\big)q_i(x,y)^{r(q_i(x,y))}, \tag{3}$$

$$q_i(x,y)=\max\{0,p_i(x,y)\}, \tag{4}$$

$$p_i(x,y)=\begin{cases} g_i(x,y), & i=1,2,\cdots,m, \\ h_{i-m}(x,y), & i=m+1,m+2,\cdots,m+l, \\ h_{i-m-l}(x,y), & i=m+l+1,m+l+2,\cdots,m+2l, \end{cases}$$

$$\big(q_i(x,y)\big)=\begin{cases} 1, & \text{若} \quad q_i(x,y)<1, \\ 2, & \text{否则}, \end{cases} \tag{5}$$

$$\theta\big(q_i(x,y)\big)=\begin{cases} 4, & \text{若} \quad q_i(x,y)<0.001, \\ 8, & \text{若} \quad 0.001\le q_i(x,y)\le 0.1, \\ 12, & \text{若} \quad 0.1<q_i(x,y)\le 1, \\ 24, & \text{否则}, \end{cases} \tag{6}$$

$$\delta(t)=t\sqrt{t}, \tag{7}$$

其中 (4) 式表示对约束的违背程度，(5) 式表示惩罚函数的强度，(6) 式为分段映射函数，(7) 式为随迭代次数变化的惩罚力度。这样根据约束违背程度的大小，自适应选择不同的惩罚项，将约束优化问题转化为一系列无约束问题来处理。

3　改进的差分进化算法

差分进化算法采用浮点数编码，在连续空间进行优化计算，是一种求解实数变量优化问题的有效方法。要将 DE 用于求解 MINP 问题，必须对 DE 进行改进。DE 的基本操作包括变异、交叉和选择操作，与其它进化算法一样也是依据适应值大小进行操作。根据 DE 算法的特点，只要对变异操作进行改进就可以将 DE 用于 MINP 问题。本文中对变异后的矢量同时向上和向下取整运算，然后择优选择变异个体进行交叉操作，从而扩大了寻优空间。对于整数变量只取 0, 1 两个变量的优化问题，在取整运算时采用四舍五入法。

3.1　变量描述与初始化

DE 由 N 个矢量 (x_i^t, y_i^t) $(i = 1,2,\cdots,N$, t 表示第 t 代) 构成种群，这 N 个矢量在搜索空间进行并行直接的寻优。设实数变量的维数为 n，整数变量维数为 d，则 (x_i^t, y_i^t) 可表示为 $(x1_i, x2_i,\cdots,xn_i, y1_i, y2_i,\cdots,yd_i)$。初始化时，根据式 (8) 和 (9) 分别对实数变量和整数变量进行初始化。对于整数变量，先在实数空间进行随机取值，然后同时向上和向下进行取整运算，择优选择其中一个个体进入初始种群，(对于 0-1 变量，先在 0, 1 空间进行随机取值，然后四舍五入取整)，得到对应的整数变量

$$x_0^i = x^L + \mathrm{rand}(\ ) * \left(x^U - x^L\right), \tag{8}$$

$$y_0^i = \left[y^L + \mathrm{rand}(\ ) * \left(y^U - y^L\right)\right], \tag{9}$$

式中 rand( ) 为 [0, 1] 之间的均匀分布随机数，[*] 表示相应的取整运算。

3.2　变异操作

DE 最基本的变异成分是父代的差分矢量，每个矢量对包括父代两个不同的个体 (x_{r1}^t, x_{r2}^t)。根据变异个体的生成方法不同，形成了多种不同的差分进化算法方案[7]，其中 DE/rand/1 方案全局搜索能力强，但收敛速度慢，DE/best/1 方案局部搜索能力强，精度高，收敛速度快，但会加大算法陷入局部最优点的可能性。结合这两种不同变异方式的特点，本文用一种新的变异方案，其变异操作方程为

$$(x_v, y_v) = \lambda(x_{r3}^t, y_{r3}^t) + (1-\lambda)(x_{\mathrm{best}}^t, y_{\mathrm{best}}^t) + F * \left((x_{r1}^t, y_{r1}^t) - (x_{r2}^t, y_{r2}^t)\right), \tag{10}$$

式中 (x_{r1}^t, y_{r1}^t), (x_{r2}^t, y_{r2}^t), (x_{r3}^t, y_{r3}^t) 为互不相同的随机个体，$(x_{\mathrm{best}}^t, y_{\mathrm{best}}^t)$ 为种群中适应值最好的个体，$F \in [0,2]$ 为缩放因子，式中 $\lambda \in [0,1]$。若 $\lambda = 1$，则 (10) 式等价 DE/rand/1 方案；若 $\lambda = 0$，则 (10) 式等价 DE/best/1 方案。一般来说，良好的算法要求在初始阶段有较强的全局搜索能力，尽可能发现多的可能全局最优点，而在后阶段则应有较强的局部搜索能力，以提高算法的精度和收敛速率。为此，将 DE/rand/1 方案和 DE/best/1 方案进行凸组合，引入线性递减加权因子 λ，将 λ 设置为退火因子，如

$$\lambda = (T_{\max} - t)/T_{\max} \tag{11}$$

所示，其中 $T_{\max}$ 表示最大迭代次数，t 表示当前迭代次数。λ 在搜索过程中由 1 逐渐变化为 0，使得 x_{r3}^t 的权重逐渐减小而 x_{best}^t 的权重逐渐增加，从而保证算法既有较强的全局搜索又有较快的收敛速率和搜索精度。

3.2.1 实数变量变异操作

$$x_v = \lambda x_{r3}^t + (1-\lambda)x_{\text{best}}^t + F*(x_{r1}^t - x_{r2}^t). \tag{12}$$

3.2.2 整数变量变异操作

对于整数变量，同时对变异矢量向上和向下取整，从而得到两个不同的变异矢量。然后对两个变异个体择优选择，进行交叉操作，这样有利于扩大搜索范围，从而有利于提高算法搜索到全局最优点的鲁棒性。对于0-1整数规划和整数只为0,1的情况，则用四舍五入的方法进行取整。对整数变量进行变异操作的方程为

$$y_v = \left[\lambda y_{r3}^t + (1-\lambda)y_{\text{best}}^t + F*\left(y_{r1}^t - y_{r2}^t\right)\right], \tag{13}$$

其中 $[*]$ 表示相应的取整运算，通过对变异矢量进行改进之后，DE就可以用于实数变量，整数变量和混合整数变量优化问题。

3.3 交叉操作

DE利用交叉操作以保持种群的多样性。对于群体中第 i 个个体 (x_i^t, y_i^t)，将与 (x_v, y_v) 进行交叉操作，产生试验个体 (x_T, y_T)。为保证个体 (x_i^t, y_i^t) 的进化，首先通过随机选择，使得 (x_T, y_T) 至少有一位由 (x_v, y_v) 贡献，而对于其它位，可利用一个交叉概率因子 CR，决定 (x_T, y_T) 中哪位由 (x_v, y_v) 贡献，哪位由 (x_i^t, y_i^t) 贡献。交叉操作的方程为

$$x_{Tj} = \begin{cases} x_{vj}, & 若 \quad \text{rand}(\ ) \le \text{R} \\ x_{ij}^t, & 否则 \end{cases} \qquad j = 1, 2, \cdots, n, \tag{14}$$

$$y_{Tj} = \begin{cases} y_{vj}, & 若 \quad \text{rand}(\ ) \le \text{R} \\ y_{ij}^t, & 否则 \end{cases} \qquad j = 1, 2, \cdots, d, \tag{15}$$

其中 rand( ) 为 $[0,1]$ 之间的均匀分布的随机数，$CR \in [0,1]$。由

$$CR = CR_{\min} + (CR_{\max} - CR_{\min})\mathrm{e}^{-a(1-t/T_{\max})^b}, \quad a = 30, \quad b = 3, \tag{16}$$

可知，CR 越大，则 (x_v, y_v) 对 (x_T, y_T) 的贡献越多，当 $CR = 1$ 时，$(x_v, y_v) = (x_T, y_T)$，有利于局部搜索和加速收敛速率；CR 越小，则 (x_i^t, y_i^t) 对 (x_T, y_T) 的贡献越多，当 $CR = 0$ 时，(x_i^t, y_i^t) 对 (x_T, y_T) 有利于保持种群的多样性和全局搜索。由此可见，在保持种群多样性与收敛速率之间是矛盾的。良好的搜索策略应该是在搜索的初始阶段保持种群多样性，进行全局搜索，尽可能得到多个可能全局最优的个体，而在搜索的后期应加强局部搜索能力，以提高算法的精度。基于这种思想，本文采用指数递增交叉概率因子 CR 的方法，即 CR 随迭代次数的增加而由小变大，初始阶段 (x_i^t, y_i^t) 对 (x_T, y_T) 贡献多，提高全局搜索能力，而在后期则 (x_v^t, y_v^t) 对 (x_T, y_T) 贡献多，提高局部搜索能力。设 $CR_{\min}$ 为最小交叉概率，$CR_{\max}$ 为最大交叉概率，t 为当前迭代次数，$T_{\max}$ 为最大迭代次数，则 CR 由(16)式确定。

3.4 选择操作

DE采用"贪婪"的搜索策略,经过变异与交叉操作后生成的试验个体 (x_T, y_T) 与 (x_i^t, y_i^t) 进行竞争，只有当 (x_T, y_T) 的适应度值较 (x_i^t, y_i^t) 更优时才被选作子代。否则，直接将 (x_i^t, y_i^t) 作为子代。选择操作的方程为

$$\left(x_i^{t+1}, y_i^{t+1}\right) = \begin{cases} (x_T, y_T), & 若 \quad F(x_T, y_T) < F(x_i^t, y_i^t), \\ (x_i^t, y_i^t), & 否则. \end{cases} \tag{17}$$

3.5 算法描述

综合以上对DE的改进，我们提出求解MINP的改进DE算法的流程如下：

步骤1 初始化种群规模N，一般为参数维数的10倍，收缩因子F，交叉概率上下界$CR_{\max}$和$CR_{\min}$，控制误差ε，在每个变量的定义域内按式(8)和(9)随机初始化每一个个体。设置差分进化算法最大迭代次数$T_{\max}$，置当前迭代计数器$t=0$，置罚函数当前迭代计数器$t=0$；

步骤2 按(4),(5)和(6)式计算每个个体每个约束条件的惩罚因子；

步骤3 按(3)式计算每个个体的所有约束条件的惩罚因子$H(x)$；

步骤4 按(2)式计算每个个体的适应值，求出最优适应值及最优个体x^t_{best}；

步骤5 判断$\delta(t)H(x,y)$是否达到精度要求，若是，则退出。否则差分进化算法开始迭代，执行下一步；

步骤6 在种群中随机选择三个与$(x_i^t,y_i^t)\,(i=1,2,\cdots,N)$不同的个体，按(12),(13)式进行变异操作，生成变异个体(x_v,y_v)；

步骤7 按(14),(15)和(16)式进行交叉操作，生成试验个体(x_T,y_T)；

步骤8 按(17)式进行选择操作，生成$t+1$代个体$(x_i^{t+1},\,y_i^{t+1})$；

步骤9 $t=t+1$，返回步骤2。

4 数值例子

为验证本文算法求解整数规划和混合整数非线性规划问题的有效性，我们采用如下四个典型测试函数进行实验[12,15,16]。

例1

$$\begin{cases}\min & f_1(x)=(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-10x_4)^4\\ \text{s.t.} & x_i\in[-100,100],\quad i=1,2,3,4,\end{cases}$$

已知最优解$(x_1,x_2,x_3,x_4,f^*)=(0,0,0,0,0)$。

例2

$$\begin{cases}\min & f_2(x,y)=2x+y\\ \text{s.t.} & 1.25-x^2-y^2\le 0,\\ & x+y\le 1.6,\\ & 0\le x\le 1.6,\\ & y\in\{0,1\},\end{cases}$$

已知最优解$(x,y,f^*)=(0.5,1,2)$。

例3

$$\begin{cases}\min & f_3(x,y)=(y_1-1)^2+(y_2-1)^2+(y_3-1)^3-\ln(y_4+1)\\ & \qquad+(x_1-1)^2+(x_1-2)^2+(x_3-3)^2\\ \text{s.t.} & y_1+y_2+y_3+x_1+x_2+x_3\le 5,\quad y_2^2+x_3^2\le 4.64,\\ & y_3^2+x_1^2+x_2^2+x_3^2\le 5.5,\quad y_2+x_2\le 1.8,\quad y_3+x_3\le 2.5,\\ & y_1+x_1\le 1.2,\quad y_2^2+x_2^2\le 1.64,\quad y_4+x_1\le 1.2,\quad y_3^2+x_3^2\le 4.25,\\ & x_1,x_2,x_3\ge 0,\quad y_1,\,y_2,\,y_3,\,y_4\in\{0,1\},\end{cases}$$

已知最优解 $(x_1, x_2, x_3, y_1, y_2, y_3, y_4, f^*) = (0.2, 1.28061, 1.95448, 1, 0, 0, 1, 3.557463)$。

例 4

$$\begin{cases} \min & f_4(x,y) = -5.357854(x_1)^2 - 0.835689y_1x_3 - 37.29329y_1 + 40792.141 \\ \text{s.t.} & a_1 + a_2y_2x_3 + a_3y_1x_2 - a_4x_1x_3 \le 92, \\ & a_5 + a_6y_2x_3 + a_7y_1y_2 + a_8x_1^2 - 90 \le 20, \\ & a_9 + a_{10}x_1x_3 + a_{11}y_1x_1 + a_{12}x_1x_2 - 20 \le 5, \\ & 27 \le x_1, x_2, x_3 \le 45, \\ & y_1 \in \{78, \cdots, 102\}, \quad y_2 \in \{33, \cdots, 45\}, \end{cases}$$

已知最优解 $(x_1, x_2, x_3, y_1, y_2, f^*) = (27, 35, 8175, 27, 78, 36, 3.22174 \times 10^4)$，至于各个参数的取值见表 1。

表 1: 例 4 中各参数的取值

a_1	85.334407	a_2	0.0056858	a_3	0.0006262	a_4	0.0022053
a_5	80.51249	a_6	0.0071317	a_7	0.0029955	a_8	0.0021213
a_9	9.300961	a_{10}	0.0047026	a_{11}	0.0012547	a_{12}	0.0019085

实验中种群规模 N 取变量维数的 10 倍，$F = 0.5$，$CR_{\min} = 0.5$，$CR_{\max} = 0.9$，例 1 至例 3 的最大迭代次数为 50，例 4 的最大迭代次数 150。为减小随机干扰，每一问题都重复 20 次实验。表 2 为每一问题的最优解，平均最优解及标准差的统计结果。

表 2: 20 次实验结果统计表

函数	已知最优解	最优解	平均值	标准差
例 1	0	0	0	0
例 2	2	2	2	0
例 3	3.557463	2.6368528	2.6368528	0
例 4	$3.221742e+004$	$3.221742e+004$	$3.221742e+004$	0

从表 2 的最优解和平均值可以看出，改进的 DE 算法在求解问题 f_3 时最优值明显好于文献 [10,17]，对于文献 [10] 的最优值为 3.557463，而本文提到的改进算法所求的最优值为 2.6368528。从标准差可以看出，改进的 DE 算法全局收敛的鲁棒性好。对于问题 $f_1 \sim f_3$，只需经过 50 次迭代运算即可得到最优解，而对于问题 f_4 也只需 150 次迭代次数，说明算法收敛速度快。图 1 为改进 DE 算法求解上述四个问题时 20 次平均最优解的进化曲线。

5 结论

采用非固定多段映射罚函数法处理问题的约束，DE 算法的变异矢量同时向上和向下进行取整运算，而对整数只包括 $\{0, 1\}$ 的混合整数非线性规划问题则用四舍五入方法进行取整运算，使改进的 DE 算法适应于求解整数规划和混合整数非线性规划问题。为保证种群的多样性和提

高算法的收敛速度，结合差分进化算法两种不同变异方式的特点，引入线性递减加权因子的凸组合变异策略，并采用指数递增交叉算子，对四个典型整数规划和混合非线性整数规划问题进行了测试。实验结果表明，改进的DE算法收敛速度快，精度高，全局搜索鲁棒性好，是一种求解混合整数非线性规划问题的有效方法，可应用于解决各种实际工程优化问题。

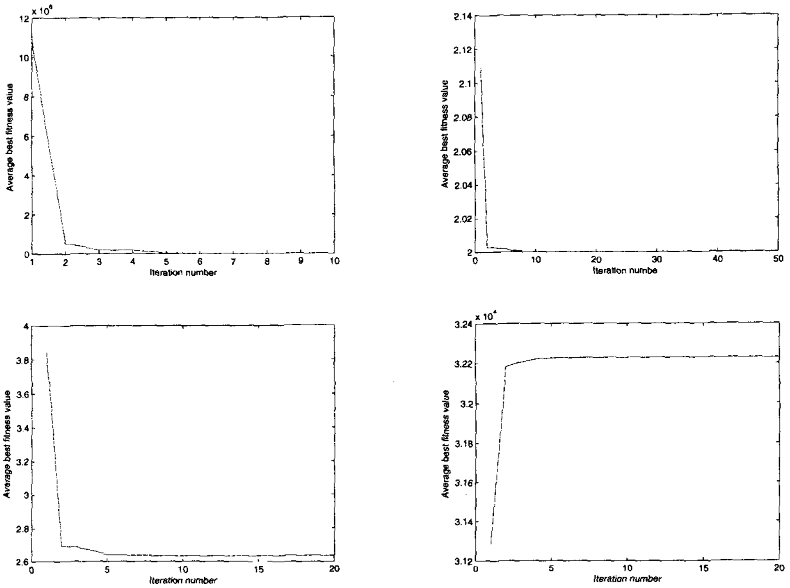

图 1: 改进DE算法求解例1至例4时20次平均最优解的进化曲线

参考文献：

[1] Wang M L, Gao X G. A *modified genetic algorithm* for solving *nonlinear hybrid optimization* problem[J]. Information and Control, 2002, 31(4): 363-366

[2] Kang Z, Li Y, Liu P, *et al*. An-all purpose evolutionary algorithm for solving nonlinear programming problems[J]. Journal of Computer Research and Development, 2002, 39(11): 1471-1477

[3] 李宏，焦永昌，张莉. 一种基于新改进的Price算法的混合遗传算法求解约束优化问题[J]. 工程数学学报，2009, 26(2): 191-199
Li H, Jiao Y C, Zhang L. Hybrid genetic algorithm based on the modification to the new version of the Price's algorithm for constrained optimization problems[J]. Chinese Journal of Engineering Mathematics, 2009, 26(2): 191-199

[4] Li L, Wang C F, Teng C X. The approximate algarithm of global optimization for a sort of nonlinear bilevel mixed integer programming problem[J]. Systems Engineering: Theory & Practice, 2002, 22(4): 19-25

[5] Liu F, Li R H. Solving mixed integer nonlinear programming problems by the evolutionary programming based on the prepotency of races[J]. Journal of System Simulation, 2003, 15(8): 1076-1078

[6] Tan Y, Gao H M, Zen J C. Particle swarm optimization for integer programming[J]. System Engineering: Theory & Practice, 2004, 24(5): 126-129

[7] Liu Z, Kang L S, Jiang L X, *et al*. New PSO algorithm for MINLP problems[J]. Mini Micro Systems, 2005, 26(6): 991-994

[8] Storn R, Price K. Differential *evolution: a simple and efficient adaptive scheme for global optimization over* continuous spaces[R]. Technical Report, International Computer Science Institute, Berkley, 1995

[9] Lin Y C, Wang F S, Huang K S. A hybrid method of evolutionary algorithms for mixed integer nonlinear optimization problems[C]// IEEE Proceeding of Evolutionary Computation Piscataway, 1999: 2159-2166

[10] 吴亮红，王耀南，陈正龙. 求解混合整数非线性规划问题的改进差分进化算法[J]. 小型微型计算机系统，2007, 28(4): 666-669

Wu L H, Wang Y N, Chen Z L. Modified differential evolution algorithm for mixed-integer nonlinear programming problems[J]. Journal of Chinese Computer Systems, 2007, 28(4): 666-669

[11] 王凌. 智能优化算法与应用[M]. 北京：清华大学出版社，2004

Wang L. Intelligent Optimization Algorithms and Applications[M]. Beijing: Tsinghua University Press, 2004

[12] 吴亮红，王耀南等. 采用非固定多段映射罚函数的非线性约束优化差分进化算法[J]. 系统工程理论与实践第，2007, 27(3): 128-133

Wu L H, Wang Y N, *et al.* Differential evolution algorithm for nonlinear constrained optimization using non-stationary multi-stage assignment penalty function[J]. Systems Engineering: Theory & Practice, 2007, 27(3): 128-133

[13] Parsopoulos K E, Vrahatis M N. Particle swarm optimization method for constrained optimization problems[C]// Intelligent Technologies-Theory and Applications: New Trends in Intelligent Technologies, Amsterdam, 2002: 214-220

[14] 曾建潮，介婧，崔志华等. 微粒群算法[M]. 北京：科学出版社，2004

Zeng J C, Jie Q, Cui Z H, *et al.* Particle Swarm Optimization[M]. Beijing: Science Press, 2004

[15] 刘芳，李人厚. 基于种族优生的进化规划用于混合非线性整数规划[J]. 系统仿真学报，2003, 15(8): 1076-1078

Liu F, Li R H. Solving mixed integer non-linear programming problems by the evolutionary programming based on the prepotency of races[J]. Acta Simulata Systematica Sinica, 2003, 15(8): 1076-1078

[16] 刘钊，康立山等. 用粒子群优化改进算法求解混合整数非线性规划问题[J]. 小型微型计算机系统，2005, 26(6): 991-994

Liu Z, Kang L S, *et al.* New PSO algorithm for MINLP problems[J]. Journal of Chinese Computer Systems, 2005, 26(6): 991-994

[17] 康卓，李艳，刘溥等. 一个通用的混合非线性规划问题的演化算法[J]. 计算机研究与发展，2002, 39(11): 1471-1477

Kang Z, Li Y, Liu P, *et al.* An all-purpose evolutionary algorithm for solving nonlinear programming problems[J]. Journal of Computer Research and Development, 2002, 39(11): 1471-1477

[18] 邓泽喜等. 一种新的差分进化算法[J]. 计算机工程与应用，2008, 44(24): 40-42

Deng Z X, *et al.* New differential evolution algorithm[J]. Computer Engineering and Applications, 2008, 44(24): 40-42

Improved Differential Evolution Algorithm for Nonlinear Mixed-integer Programming Problems

LIU Jun-mei[1,2], GAO Yue-lin[2]

(1- Department of Basic Mathematics, Yinchuan College, China University of Mining and Technology, Yinchuan 750011; 2- Institute of Information and System Science, North National University, Yinchuan 750021)

Abstract: For nonlinear mixed-integer programming problems, we use the non-stationary multi-stage mapping penalty function to handle constraints and use mixed integer coding technology to handle real-valued variables and integer-valued variables. At the same time, a new convex combination mutation operator and an exponent increased crossover probability operator are integrated into the differential evolution algorithm. We construct an improved differential evolution algorithm for nonlinear mixed-integer programming problems. The experiments show that the proposed algorithm has fast global convergence speed, high computational precision and good robustness.

Keywords: global optimization; nonlinear mixed-integer programming; non-stationary multi-stage mapping penalty function; differential evolution algorithm

Received: 26 Dec 2008. **Accepted:** 07 Sep 2010.

Foundation item: The National Natural Science Foundation of China (60962006).